Using the QMAX 650 CCD Camera at MHATT/XOR

Dohn Alexander Arms
dohnarms@anl.gov

May 26, 2005

1 Introduction

The purpose of this document is to help people use the one-of-a-kind QMAX 650 CCD camera at MHATT/XOR. This camera is provided for our use by Dr. Roy Clarke of the University of Michigan. Fig. 1 shows it *in situ*, as used in indirect mode.

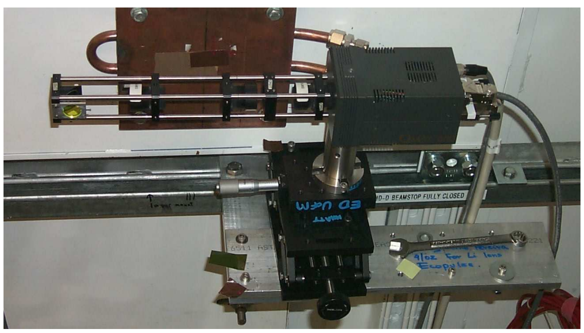

Figure 1: A picture of the CCD camera, in indirect mode.

The camera uses the Texas Instruments TC237B chip, which currently costs around \$100. The chip has 680×500 pixels of 7.4 μm square. The full specification sheet for this chip is available at TI's web site [1]. The CCD chip in the camera is easily replaceable, in case one gets damaged.

The chip can be used to look at X-rays in two different modes: direct X-ray detection and indirect fluorescence detection. For direct detection, the glass that covers the active area of the chip absorbs X-rays, and it needs to be removed (which is an imprecise affair which can damage the chip). While used for direct detection, the chip can become damaged over time, especially from high energy photons, which eventually requires that the chip be replaced. For indirect fluorescence detection, the X-rays strike a thin fluorescent crystal (in our case a cerium doped YAG crystal; it's the yellow crystal on the left in Fig. 1), with the resulting fluorescence imaged through a mirror by lenses onto the CCD chip with 1:1 magnification [2]; as the glass on the CCD chip is transparent to this light, an unaltered chip is used in the camera for this mode.

The camera returns a 12-bit 640×467 image to the computer.

2 Connecting the cables

The camera has three cables that need to be hooked up to it, as shown in Fig. 2. The camera is hooked up to a PC running Windows 98. Two of the cables have to go to the computer, being the data and serial cables, while the power cable can connect to the computer, but doesn't have to. The one normally used at MHATT/XOR for this purpose is our computer-on-a-cart, named 'wavelet'.

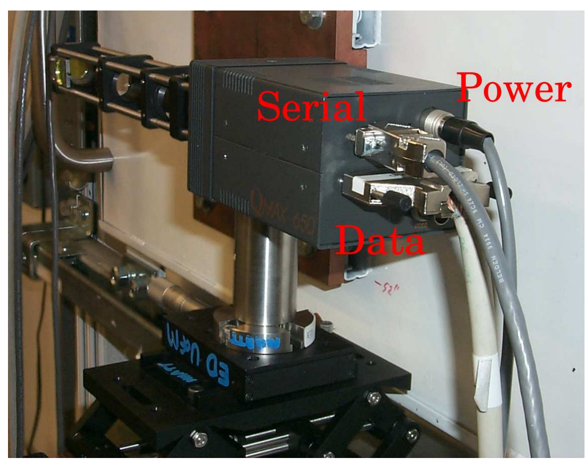

Figure 2: A picture of the back of the camera, with all three cables in place.

The data cable resembles a very long 68-pin SCSI-2 cable, and connects to a CCD grabber board installed in the computer. Currently, it's in fragile condition, so care has to be taken when using it. It's over 50 feet long, and it connects directly to the camera and computer, through labyrinths in the hutch walls. The cable end with thumb-screws is normally connected to the camera, and the end with no screws is connected to the computer. The computer end sometimes needs to be jiggled a little to get a good connection; hopefully this will be fixed in the near future. The data cable as it connects to the computer is shown in Fig 3.

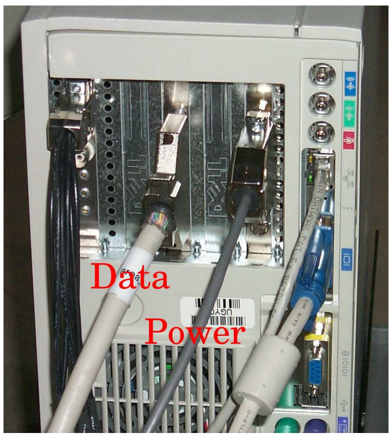

Figure 3: A picture of the back of the computer, with data and power cables in place. The serial cable is connected to COM1 below and off-picture.

A 9-pin serial connection is needed for the camera, which can be done through a patch panel. However, if a direct connection over a long distance is needed, there is a gray 50 foot long cable with DB-9 connectors of opposite sexes, which was made for this purpose. , normally connected directly between the camera and the computer. The camera has a female connection, while the computer has a male connection. The cable can be connected into either COM1 or COM2 on the computer, but COM1 is more convenient software-wise.

The camera uses needs only a 12 V power source, but it draws quite a bit of current. The camera initially draws around 1.5 A when plugged in, after which it then draws 0.8 A. There are two choices for connecting the camera: directly to the computer or to a power supply. This choice is accomplished by the fact that the power cable from the camera has a black AMP 9-pin connector on the other end, with corresponding mating cables to whichever of the two options used.

To get the camera's power from the computer, the long cable with a male DB-15 connector on the end is used. On the computer, there is a card which supplies the power and has a female DB-15 connector. The drawback of this way is that this cable is yet another to string through the labyrinth. The power cable as it connects to the computer is shown in Fig 3.

Figure 4: A picture of the unconnected power cable for use with the computer.

To get the camera's power from a power supply, the short cable that also has a black AMP 4-pin connector on it is used; there is a plastic box in the middle, with orange tape around it for easy identification. The 4-pin AMP connector is plugged into a special power supply, as shown in Fig. 5, which delivers the needed power.

There are banana jacks on the front that one can used, even while the camera is running. Care should be taken however, as the camera draws between 0.8 and 1.2 A using the 12 V source. The supply should be considered to have a maximum current of 2.5 A at 5 V and 0.6 A at 12 V while the camera is drawing power from it.

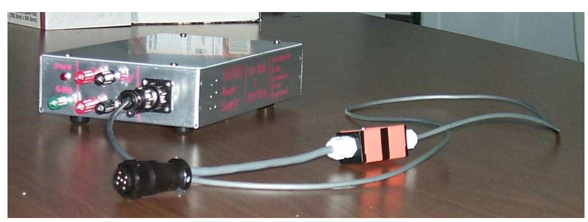

Figure 5: A picture of the power supply and cable end.

3 Mounting the camera

In order to mount the camera, we suggest using a 1" diameter optical mount that has a 1/4-20 set screw in the end. The set screw can be screwed into the screw hole on the bottom of the camera near the front of the camera. This is how the camera is mounted in Fig. 6. The post is then mounted directly onto an optical or some other type of mount. Whatever the mount, make sure it is very rigid; the camera is front-heavy (due to a copper mask) and the cables can exert quite bit of force (especially if someone trips on them).

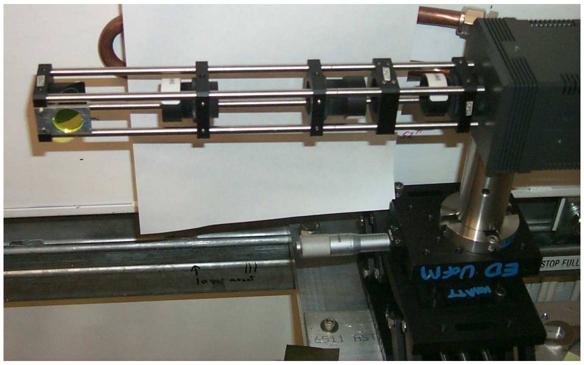

Figure 6: A close-up of the front of the camera, while in indirect mode.

Normally, the camera needs to be moved side-to-side and up-and-down to get the beam into the detection area of the CCD chip. Part of the mount for the camera should have some easy way to move the camera in these directions. If the camera does not need to be moved once the beam is found, using manually cranked stages, jacks, or slides is the best solution. If the experiment requires moving the camera while taking data, then motorized versions of these devices are needed.

If the camera is be used in direct-detection mode, the CCD chip needs to have the glass removed. No lenses are mounted on to the camera, although a lead tape mask is normally used to block off everything but the active area of the chip. The way the camera is set up in this case is to face it towards the incident X-ray beam.

In indirect mode, the chip can have glass on it or not. If the chip had previously exposed to direct beam, it may have damage, which might require a change of chip. The lens jig needs to be put onto the front of the camera, like in Fig. 6. The yellow YAG crystal is faced towards the direction of the incident X-ray beam. When the YAG fluoresces, the illumination will reflect off of the 45 ° mirror, through a lens, and focus onto the CCD chip. The lens does need to be focussed. When the viewing software is running, the lens needs to be moved so that the mark on the front of the YAG is in focus. The lens is moved by loosening a retaining ring, screwing in or out the lens, and then tightening the retaining ring again.

4 KSA 400 software

The QMAX 650 has an external control program that needs to be run before starting the KSA software. The name of the program is `QMax650.exe`, and its icon is normally left on the Windows desktop. Run it, and it will open to a window. If the serial cable is not connected to the `COM1` port, change the *COM Port* selection to the correct value. The other settings need to be left at their defaults. Do not close this window, as it is needed while running the camera. Change the *Exposure Time* to a new value, which forces the software to talk to the camera (if you don't do this, KSA 400 will complain later).

Start the KSA 400 software. From the **View** menu, select **One Live Video**. Once the window is open, from the **Edit** menu, select **Properties**. Select the **Camera** tab, and in the *Camera* field, select *QMax 650*, then hit *Apply* and close the window.

Change the *Exposure Time* again until you see something in the video window of KSA 400, maybe 40 msec. This software is weird because any changes you make happen *immediately*, so if you delete a value, the program complains because it needs a value, but nothing bad happens.

At this point, you should have a live video feed from the camera. To actually take data, you need to select a mode from under the **Acquire** menu (only one can be selected at a time). **Single Image** is the most basic, and grabs one frame. **Multiple Images** will take a series of images one after another, separated by some time delay, saving them into individual files. **Movie** will take multiple frames in sequence as well, but puts them into a single file. **Focus Mode** will open a seperate window that displays the cross section of the camera output, selected on the video window with a movable line; multiple lines can be used at the same time. **Interactive Accumulation** will take multiple grabs of the camera until stopped, summing together into one image file.

Once the data has been acquired, it needs to be saved, which is not automatic. You need to click on the image you want saved, then under the **File** menu, select **Save As** and save it to a directory clearly showing whose files these are.

5 Data file format

The KSA software saves an image (single frame) or an accumulation image (summed frame) into files with a **.img** ending, and saves an uncompressed movie (multiple frames) into files with a **.imm** ending. These formats are proprietary and there are no published documents for it. In principle, the header inside the file needs to be read in order to know what the parameters of the images inside really are. However, if the images or movies were taken using the QMAX 650 camera, the header can be ignored and the data extracted.

First we consider the simplest type of file, the single frame image. There is a header of 640 bytes at the beginning of the file. The camera sends back 12 bits per pixel, but each pixel is stored in 16 bits on the computer, thus each pixel uses 2 bytes. Since the dimensions of a frame are 640×467, the total size used for the image is $640 + 2 \times 640 \times 467 = 598\,400$ bytes, which is the same as the file size.

Next we consider the uncompressed movie file, which has a number of frames inside it. The format actually is very simple, having the single-image format repeated over and over. There is a header, then data, then another header, then more data, etc. The file size ends up being a multiple of 598 400, which can be used to tell how many frames there are in a file.

An accumulation frame image is a little different in make-up. Since many frames will be summed together on the computer, the computer stores each pixel sum in 32 bits (4 bytes). The header size is the same 640 bytes as before. The file size is thus $640 + 4 \times 640 \times 467 = 1\,196\,160$ bytes.

The image data in the file is written with *little endian* byte order (ironically, this is backwards from the way the header is written). PC's are *little endian*, while SUN's are *big endian*. What this means is that the lowest byte of data, B_1, has the "least important" imformation, while the highest byte of data, B_N, has the "most important" information for N pixels/data. If we assume 4 byte/pixel data, this means the pixel value is

$$\text{pixel} = B_1 + 2^8 B_2 + 2^{16} B_3 + 2^{24} B_4 \,.$$

In C code, if one has

```
unsigned char b[4];
unsigned long int pixel;
int i;
```

and a 4 byte set of data is loaded into `b[4]`, then

```
pixel = 0;
for( i = 0; i < 4; i++)
  pixel += (((unsigned long int) b[i]) << 8*i);
```

loads the pixel value into `pixel`. If you are using a *little endian* computer, `fread` can also be used to directly load the image into memory.

6 Data file conversion tools

For those who do not wish to manipulate the data in the KSA format, there is software written at MHATT/XOR for conversion to other formats.

On 'hydra' is a program `ksa2pgm` that converts an **.img** file to a PGM file, which is a very general ASCII greyscale format common on UNIX platforms. The program does however scale an image to fit into 16 bits/pixel. The files generated are quite large, but PGM files are useful for passing into another filter, converting the PGM file into yet another format (the *netpbm* tools have such filters [3]).

On 'hydra' is another program `ksa2png` which does a conversion to the PNG format. This actually is a script which calls `ksa2pgm` and sends the output through `pnmtopng`, making a greyscale PNG file.

References

[1] http://www-s.ti.com/sc/psheets/socs063/socs063.pdf

[2] Described in an APS 2000 activity report by Dufresne and Dierker, available at http://www.aps.anl.gov/apsar2000/dufresnee2.pdf

[3] http://sourceforge.net/projects/netpbm/